

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-106337

(43)Date of publication of application : 22.04.1997

(51)Int Cl.

G06F 3/14

G06F 9/06

(21)Application number : 07-289401

(71)Applicant : OKI ELECTRIC IND CO LTD

(22)Date of filing : 11.10.1995

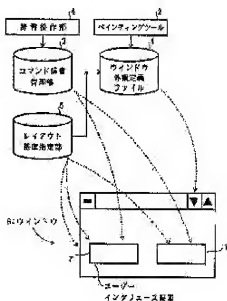
(72)Inventor : MAEDA TADAHIKO

(54) USER INTERFACE GENERATOR

(57)Abstract:

PROBLEM TO BE SOLVED: To preliminarily prepare the stylized shaped user interface element for a command to be generally used, to stick the element by fixed layout on a stylized window, to facilitate the design of a new window, to always generate the standard window without unnatural feeling and to improve the productivity of software developments.

SOLUTION: A window appearance definition file 1 defines a window 6 in which a preliminarily set and prescribed design is stylized. A command dictionary control part 3 stores and holds the prescribed stylized attributes corresponding to commands to be generally used and designates the shape of a user interface element 7 such as a command button, etc., to be displayed on the window. A layout reference designation part 5 stylizes the arrangement and shape, etc., of the command button and designates them.



特開平9-106337

(43) 公開日 平成9年(1997)4月22日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 3/14 9/06	3 7 0 5 3 0		G 0 6 F 3/14 9/06	3 7 0 A 5 3 0 N

審査請求 未請求 請求項の数 6 F D (全 12 頁)

(21) 出願番号 特願平7-289401

(22) 出願日 平成7年(1995)10月11日

(71) 出願人 000000295

沖電気工業株式会社
東京都港区虎ノ門1丁目7番12号

(72) 発明者 前田 忠彦

東京都港区虎ノ門1丁目7番12号 沖電気
工業株式会社内

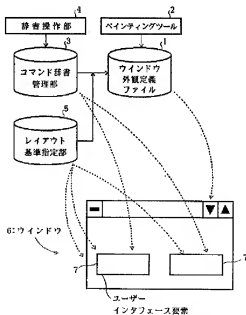
(74) 代理人 弁理士 佐藤 幸男 (外1名)

(54) 【発明の名称】 ユーザーインタフェース生成装置

(57) 【要約】

【解決手段】 ウィンドウ外観定義ファイル1は、予め設定した所定のデザインの定型化されたウィンドウ6を定義する。コマンド辞書管理部3は、汎用されるコマンドに対応する所定の定型化された属性等を格納して保持し、ウィンドウ上に表示するコマンドボタン等のユーザーインタフェース要素7の形状等を指定する。レイアウト基準指定部5はコマンドボタンの配置や形状等を定型化して指定する。

【効果】 汎用されるコマンドに対しては定型化された形状のユーザーインタフェース要素が予め用意され、しかも定型化されたウィンドウ上に一定のレイアウトで貼り付けられることができるため、新たなウィンドウの設計が容易になり、常に標準的な違和感の無いウィンドウが生成され、ソフトウェア開発の生産性が向上する。



本発明のユーザーインタフェース生成装置概略図（その1）

1

【特許請求の範囲】

【請求項1】 情報処理装置の画面に表示されるウィンドウ状のグラフィカルユーザーインターフェースを生成するものであって、

予め個々のウィンドウ中に表示される定型のユーザーインターフェース要素とそのコマンドの内容に応じた属性と形状とを対応付けて辞書化したコマンド辞書管理部と、定型のウィンドウの形状を指定するウィンドウ外観定義ファイルと、

このウィンドウの中に表示される定型のユーザーインターフェース要素の配置を指定するレイアウト基準指定部とを備えたことを特徴とするユーザーインターフェース生成装置。

【請求項2】 情報処理装置の画面に表示されたあるウィンドウから別のウィンドウに遷移するためのウィンドウの遷移過程を選択すると、その遷移動作を命令するためのユーザーインターフェース要素のウィンドウ上の配置や形状と挙動を定型化したウィンドウを生成するダイアログプログラム指定部を備えたことを特徴とする請求項1記載のユーザーインターフェース生成装置。

【請求項3】 所定の関連性のある命令を入力するための複数のユーザーインターフェース要素をグループ化して、一括登録を可能とするコマンドグループ指定部を備えたことを特徴とする請求項1記載のユーザーインターフェース生成装置。

【請求項4】 メニュー階層構造を表示し、その階層に従ってプログラム制御をする一連のコマンド群の一括登録を可能とするコマンドグループ指定部を備えたことを特徴とする請求項1記載のユーザーインターフェース生成装置。

【請求項5】 コマンド辞書管理部は、コマンド名に応じたボタンの並び順を設定するためのウィンドウ上に配列された各ボタンの位置情報を表示することを特徴とする請求項1記載のユーザーインターフェース生成装置。

【請求項6】 コマンド辞書管理部は、コマンド名に応じたボタンのデフォルト設定を表示することを特徴とする請求項1記載のユーザーインターフェース生成装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、パーソナルコンピュータやワークステーションの画面に表示されるウィンドウ設計のためのユーザーインターフェース生成装置に関する。

【0002】

【従来の技術】 パーソナルコンピュータやワークステーションにおいては、各種のアプリケーションプログラムを起動し操作するために画面上に各種のウィンドウが表示される。このウィンドウ上には、コマンドを入力するためのコマンドボタンやリストボックス、その他各種のユーザーインターフェース要素が表示され、ユーザーによ

2

る操作性の向上を図っている。このようなウィンドウ等から構成されるユーザーインターフェースの開発は、従来、プログラマーが各種のウィンドウの背景をデザインし、設計生成するようにしていた。

【0003】 この場合に、ウィンドウ生成用のツールを用いて、プログラマーはそのウィンドウの背景をデザインし、各種のアイコンやコマンドボタン、その他のユーザーインターフェース要素の位置や大きさを決定して、これを貼り付け、更に各コマンドに対応する処理の内容を個別にコーディングして登録するといった処理を行っていた。例えばコマンド入力用のボタンの場合には、それぞれその大きさや位置等のプロパティをセットしたり、あるいは必要に応じて既存のユーザーインターフェースに採用しているコマンドのデザイン等をそっくりコピーして貼り付けるといった方法が採用されていた。

【0004】

【発明が解決しようとする課題】 ところで、上記のような従来使用されているユーザーインターフェース生成装置には次のような解決すべき課題があった。ユーザーインターフェースにはファイル入出力処理のような極めて定型的な処理を行うものがある。このようなウィンドウは全体として予め用意され、一括してコピーして利用することもできる。ところが、アプリケーションソフトウェアには、多様な処理に対応する多くの固有のウィンドウが要求され、そのデザインや設計はプログラマーに対して大きな負担となっていた。しかし、各ウィンドウには部分的に定型的な処理を行うものが含まれていることも多い。例えば、一定の処理を開始する前には、ウィンドウ上にメッセージを表示してその了解を得る処理がある。この場合、OKボタンとキャンセルボタンとを表示する。

【0005】 こういった場合に、プログラマーによってウィンドウやボタンのデザインが異なると、ユーザーに対し違和感を与え、統一された環境下での操作性を損なうおそれがある。例えば、OKボタンとキャンセルボタンとが左右に並んでいる場合、どちらを左にどちらを右にするかあるいはその大きさや配置をどうするかといった問題は可能な限りどのアプリケーションも共通化しておきたい。

【0006】

【課題を解決するための手段】 本発明は以上の点を解決するための構成を採用する。

〈構成1〉 情報処理装置の画面に表示されるウィンドウ状のグラフィカルユーザーインターフェースを生成するものであって、予め個々のウィンドウ中に表示される定型のユーザーインターフェース要素とそのコマンドの内容に応じた属性と形状とを対応付けて辞書化したコマンド辞書管理部と、定型のウィンドウの形状を指定するウィンドウ外観定義ファイルと、このウィンドウの中に表示される定型のユーザーインターフェース要素の配置を指定す

3

るレイアウト基準指定部とを備える。

【0007】〈説明〉ユーザインタフェース要素とは、ウィンドウ上に設けられ、例えばマウスでクリックして処理を起動するためのコマンドボタン等をいう。定型のとしたのは、標準的・一定の形状のユーザインタフェース要素を予め用意することを意味する。ユーザインタフェース要素とそのコマンドの内容に応じた属性と形状とを対応付けて辞書化すると、そのユーザインタフェース要素を指定すれば、辞書化により、自動的に標準的なコマンドボタン等がウィンドウ上に登録できる。ウィンドウ外親定義ファイルによって、定型化されたウィンドウを表示し、レイアウト基準指定部によって、ウィンドウ上のユーザインタフェース要素の配置も含めて定型フォーム化すれば、ウィンドウの設計が容易になる。また、同様の処理を実行するためのウィンドウが作成者によってまちまちなデザインにならないで統一化され、ソフトウェア開発の生産性も向上する。

【0008】〈構成2〉情報処理装置の画面に表示されたあるウィンドウから別のウィンドウに遷移するためのウィンドウの遷移過程を選択すると、その遷移動作を命令するためのユーザインタフェース要素のウィンドウ上の配置や形状と挙動を定型化したウィンドウを生成するダイアログプロ—指定部を備える。

〈説明〉あるウィンドウから別のウィンドウに遷移する処理を実行させるには、遷移前後のウィンドウと遷移イベントを起動させるためのユーザインタフェース要素の設定や、遷移動作等を登録しなければならないが、このウィンドウの遷移動作やそのウィンドウのデザインも含めてダイアログプロ—指定部に定型化して登録しておけば、ウィンドウ遷移に伴う処理プログラムの設計も省力化できる。

【0009】〈構成3〉所定の関連性のある命令を入力するための複数のユーザインタフェース要素をグループ化して、一括登録を可能とするコマンドグループ指定部を備える。

〈説明〉例えば処理の開始の際に表示されるメッセージウィンドウには、「OK」と「キャンセル」のいずれかを選択する一対のボタンが使用される。このようなボタンは、個別に登録するのではなく、グループ化して一括登録できれば、プログラム設計が容易になる。

【0010】〈構成4〉メニュー階層構造を表示し、その階層に従ってプログラム制御をする一連のコマンド群の一括登録を可能とするコマンドグループ指定部を備える。

〈説明〉階層化されたメニューも定型的なものはグループ化して一括登録を可能にした。

【0011】〈構成5〉コマンド辞書管理部は、コマンド名に応じたボタンの並び順を設定するためのウィンドウ上に配列された各ボタンの位置情報を表示する。

〈説明〉コマンドボタンの性質に応じてその並び順も定

4

型化しておけば、ウィンドウのデザインがより統一される。

【0012】〈構成6〉コマンド辞書管理部は、コマンド名に応じたボタンのデフォルト設定を表示する。

〈説明〉デフォルトボタンの設定も定型化しておけば、ユーザインタフェース設計作業がより省力化できる。

【0013】

【発明の実施の形態】以下、本発明の実施の形態を具体例を用いて説明する。

10 〈具体例1〉図1は、本発明のユーザインタフェース生成装置具体例を示す概略図である。図に示すように、この装置はウィンドウ外親定義ファイル1、コマンド辞書管理部3、レイアウト基準規定部5等が所定の記憶部に設けられたもので、ウィンドウ外親定義ファイル1の内容を操作するために、ペインティングツール2が設けられ、コマンド辞書管理部3の内容を操作するために辞書操作部4が設けられたものである。このような装置によって、図の下側に示すようなユーザインタフェースであるウィンドウ6が生成される。このウィンドウ6には、例えばコマンドボタンのようなユーザインタフェース要素7が、適当な数、適当な位置に配置されている。

【0014】ウィンドウ外親定義ファイル1は、定型的な標準的なウィンドウ6の外観、即ち大きさや背景等を定義し格納しておくファイルである。ここにはウィンドウの種類に応じてそれぞれ所定の外親定義を行った情報が格納されている。このようなウィンドウは、グラフィック処理を行うためのアプリケーション等から構成されるペインティングツール2によって自由に生成され、予め任意の数だけ格納されている。コマンド辞書管理部3は、各ユーザインタフェース要素7のコマンドの内容に応じた属性等を規定したものである。

【0015】図2に、このようなコマンド辞書の内容説明図を示す。この図に示すように、コマンド辞書にはコマンド名とそのコマンドの種別及び属性等が格納されている。例えば、コマンド名が「了解」という名の命令であればコマンド種別は「プッシュボタン」とされ、そのプッシュボタン上に表示される文言、即ちキャプションは「OK」という内容になる。また、大きさも定型化されており、ここでは300×500（ドット）という構成になっている。

【0016】図1に示すレイアウト基準規定部5は、ウィンドウ6の上に貼り付けられたいくつかのユーザインタフェース要素7の配置等を指定する。図3には、このレイアウト基準の内容説明図を示す。例えば、この例では、ユーザインタフェース要素7がプッシュボタンである場合には、その配置は左下から右下へ等間隔に行うといった基準になっている。従って、ウィンドウ上にいくつかのプッシュボタンが貼り付けられる場合には、この基準に従って自動的にウィンドウ6の下の部分にボ

5

タンが等間隔に1列に並べられる。

【0017】次に、図1に示すユーザーインタフェース生成装置の具体的な動作を説明する。なお、この装置を動作させるために、まず生成すべきウィンドウと、ここに貼り付けるユーザーインタフェース要素を指定するための情報を作る。図4には、ウィンドウ名とコマンドの対応表説明図を示す。一定のウィンドウを表示させ、このウィンドウによって各種のコマンドを実行させるために、ウィンドウ名とコマンド名とを対応付けて、この図に示すように表示する。

【0018】図5は、図1に示す装置のユーザーインタフェース生成動作フローチャートである。まず、ステップS1において、図4に示したウィンドウとコマンドとの対応表を参照し、ウィンドウ名を指定すると共にそのウィンドウに貼り付けるコマンド名を調べる。そして、ステップS2において、順にそのコマンド名を取り出し、ステップS3において、そのコマンド名はユーザーインタフェース要素と対応付けられるかどうかを判断する。ユーザーインタフェース要素をウィンドウ上に貼り付ける場合にはステップS4に移り、図1に示すコマンド辞書管理部3とレイアウト基準指定部5を参照してユーザーインタフェース要素を追加する。即ち、図2のコマンド辞書によりコマンド名を指定すれば、自動的にそのコマンド種別や属性等が決定される。また、図3のレイアウト基準によりそのウィンドウ上の配置も決定される。そして、このようなデータを保持し、更にステップS1に戻り、次に処理すべきコマンドがあるかどうかを判断する。例えば、図4に示した例では、ウィンドウW1に貼り付けるべきコマンド名として2種のコマンドが挙げられている。従って、図5の処理はステップS1～S4まで2回繰り返される。なお、いずれかのコマンドがユーザーインタフェース要素を要しないものであれば、ステップS4は実行しない。

【0019】こうして、全てのコマンドについて必要な情報の参照が終了と、ステップS1からステップS5に進み、ウィンドウ外観定義ファイル1を参照してウィンドウ外観に関する情報を取り出す。そして、ステップS4において保持しておいたデータを元にして、ウィンドウ上に必要なコマンドに対応するコマンドボタン、その他のユーザーインタフェース要素を貼り付ける。こうして、この具体例によるユーザーインタフェースの生成が完了する。

【0020】〈具体例1の効果〉以上の具体例によれば、プログラマーは、予めウィンドウの種類とそのウィンドウ上に貼り付けるべきユーザーインタフェース要素に対するコマンド名のリストを指定すると、自動的に必要なコマンドに対応するコマンドボタン等を貼り付けたウィンドウが生成される。この場合に、プログラマーはコマンドボタンの属性や大きさや配置等を個別に指定し入力するといった作業を行わないで済む。即ち、従来よ

6

りも容易にユーザーインタフェースの生成が可能となる。しかも、こうして生成されたユーザーインタフェースは、ボタンの形状、配置、大きさといった外観の主要要素が定型化されているため、アプリケーションプログラムのユーザーにとっても違和感の無い操作性のよいものとなる。

【0021】〈具体例2〉図6に、具体例2による本発明のユーザーインタフェース生成装置概略図を示す。この装置には、図1の装置に更にダイアログフロー指定部8及び挙動定義ファイル9を追加したものである。ユーザーインタフェースとして画面上に表示されるウィンドウを利用するアプリケーションプログラムでは、一定の処理を実行する場合には一定のウィンドウが表示され、そのウィンドウ上で他の一定の処理を実行する指示を行うと、ウィンドウが切り換えられるといったウィンドウの遷移が頻繁に行われる。これをダイアログフローという。この場合のウィンドウの遷移をさせるための挙動を、コマンドに対応して予め定型化しておくことによって、ウィンドウ遷移を含めたユーザーインタフェースの生成処理が簡便になる。この具体例はそのような目的を達成する。

【0022】図7には、ダイアログフローの内容説明図を示す。例えば、この図に示すように始めにウィンドウW1が表示されているものとする。このとき、ウィンドウW1上の“開く”という内容のコマンドボタンを押しした場合には、ウィンドウW2が開き、ウィンドウW2上の“詳細”というコマンドボタンを押しした場合にはウィンドウW3が開く。そして、ウィンドウW3上の“閉じる”というコマンドボタンを押しした場合、再びウィンドウW2に戻り、ウィンドウW2上の“閉じる”というコマンドボタンを押しすると、ウィンドウW1に戻る。また、ウィンドウW2上の“参照”というコマンドボタンを押しすると、ウィンドウW4が表示され、ウィンドウW4上の“閉じる”というコマンドボタンを押しするとウィンドウW2が表示される。また、ウィンドウW4上の“反映”というボタンが押しされると、ここではウィンドウW4を表示したまま一定の内部処理が実行される。なお、“開く”、“参照”、“詳細”といった実線で示した矢印は、これまで表示されていたウィンドウに重ねて別のウィンドウを開く追加表示を指示する処理を表している。このような矢印をここでは、遷移線と呼んでいる。また、破線の矢印で表した遷移線は、これまで表示していたウィンドウを消してから元のウィンドウに戻る処理を表している。この他に、ウィンドウW4の“反映”という処理を表するループ状の遷移線は内部処理を表している。

【0023】図8には、ダイアログフロー指定部8を利用した本発明の装置の動作フローチャートを示す。まず、この処理を行う場合には、ダイアログフロー指定部8に

7

記憶された図7に示すようなダイアログフローから順に遷移線を取り出す。そして、そのウィンドウの遷移を処理させるコマンド名をステップS2において取り出す。このコマンド名がユーザーインタフェース要素となるかどうかをステップS4において判断し、ユーザーインタフェース要素であればステップS4において、既に図5を用いて説明したと同様に、コマンド辞書管理部3とレイアウト基準指定部5の参照によりユーザーインタフェース要素を追加する。

【0024】次に、ステップS5において、遷移線の種別とコマンド名とからコールバック処理とファイルイベント処理を追加する。即ち、具体例1で説明したようなコマンドの外観のみならず、そのコマンドを実行する際のウィンドウ遷移を発生する処理の挙動定義を追加する。ここで、コールバック処理というのは、一定のコマンドが発生した場合の挙動を定義したものである。例えば、「開く」という内容のコマンドの種類がメニュー及びプッシュボタンであったとする。このとき、そのコマンドボタンをプッシュするイベントが発生すると、ウィンドウW1を表示していた状態からその上にウィンドウW2を追加表示する遷移が発生するものとすれば、その挙動をコールバック処理の内容として定義する。具体的には、そのコマンドボタンに対して、ウィンドウW1に対しては何の処理もせず、ウィンドウW2を重ねて追加表示するといった内容の処理をコーディングして登録する。

【0025】一方、内部イベント処理というのは、あるウィンドウ上での内部イベントを発生する処理、例えばコンボボックス中のデータを参照するような場合の処理をいう。従って、例えば「参照」というコマンドに対しては、図7に示したようなウィンドウW4に対する定型処理を生成する。

【0026】このようにして、図7に示したダイアログフローの各遷移線に対応するコマンドについて、その挙動定義を全て行った後、図8のステップS1からステップS5に移る。そして、ステップS4で行ったウィンドウの外観定義とステップS5で行った挙動定義とを併せて出力する。

【0027】〈具体例2の効果〉以上の具体例によって、あるウィンドウから他のウィンドウへの遷移を実行させるユーザーインタフェース要素が存在する場合に、その外観や配置等の定義のみならず、他のウィンドウに対しどのような条件で遷移するかといった挙動を定型化して、図6に示す挙動定義ファイル9に格納しておくことにより、ウィンドウ遷移の制御も簡便な処理で定型化されたものが生成される。なお、上記のような挙動定義は、図6に示す挙動定義ツール10を用いて自由に新たに追加し変更することも可能である。

【0028】〈具体例3〉ウィンドウW1上に貼り付けられるコマンドは単独で使用される場合の他、2種あるいは

8

3種類以上、グループで使用されることも多い。例えば、一定の処理について、ユーザーに確認を求める場合には、了解ボタンと取消しボタンとが対になって使用される。このように一般的にいくつかのボタンが一括して使用されるような場合に、一括して一挙にウィンドウ上に貼り付けられるようにすれば、ウィンドウの生成処理がより簡便になる。この具体例3は、そのようなウィンドウ生成を実現している。

【0029】図9は、グループ機能付きコマンド辞書説明図である。この具体例の場合には、図1に示すコマンド辞書管理部3に、この図に示すような内容のデータを格納する。この図9（a）には、コマンドグループ名とそのグループを構成するコマンド名の例を対応付けて示した。また、（b）には、各コマンド名に対応するコマンド種別や属性等を表示した。これは、既に図2を用いて説明したものと同様の内容である。このコマンド名は、コマンドグループ名に対応する個々の構成コマンドに対して設けられたものである。即ち、コマンドグループ名が「確認」という内容のものであれば、これは了解と取消しというコマンドの対によって構成される。また、コマンドグループ名が「取消し」というものであるれば、「再試行」と「取消し終了」といった一対のボタンで構成される。即ち、プログラム開発者が予め各コマンドを指定するのでなく、コマンドグループ名を指定することによって一挙にいくつかのコマンドのウィンドウへの登録が可能となる。

【0030】図10には、コマンドグループを利用したダイアログフローの説明図を示す。この図に示す例では、ウィンドウがW1～W4まで遷移している場合には、それぞれの遷移線に一定のコマンド、あるいはコマンドグループ名等が対応付けられ表示されている。まず、ウィンドウW1からウィンドウW2への遷移線に対応するコマンドは、グループコマンドである「確認」という処理の中の了解という処理である。また、ウィンドウW1からウィンドウW4への遷移コマンドは「確認」という内容のグループコマンドのうちの「取消し」というグループコマンド中の「再試行」というコマンドによる。なお、ウィンドウW1からの遷移動作として、確認/取消し/取消し終了というコマンドが存在しない。これは、このコマンドに対する遷移が内部処理、即ちデフォルトの処理だからである。即ち、このときはウィンドウの遷移が発生しない。ウィンドウW2からウィンドウW3への遷移線に対応するコマンドは「確認」である。これは、確認/了解、確認/取消し再試行、確認/取消し/取消し終了の全てのコマンドに対してこのような遷移が発生することを意味する。従って、ウィンドウW2には「了解」と「取消し」及び「取消し終了」の各コマンドボタンが表示され、これらの操作によっていずれもウィンドウW3への遷移が発生する。

【0031】本発明の装置は、以上のようなウィンドウ

遷移を発生するための各コマンドについて、先に図8を用いて説明したようなウィンドウの外観定義や挙動定義を自動的に行う。図11は、この目的のためにコマンドグループを使用したダイアログフロー指定部利用の動作フローチャートである。図11のステップS1において、図10に示したようなダイアログフローを参照して次の遷移線があるかどうか判断を行う。これは、図8のステップS1で行った処理と同様である。また、ステップS2も、図8のステップS2と同様に遷移線に対応するコマンド名の取出しを行う。そして、ステップS3において、グループコマンドかどうかを判断する。グループコマンドであればステップS4に進み、下位のコマンド名を取り出す。この下位のコマンドは、図9(a)に示したようなコマンド辞書を参照することによって取り出される。ステップS5では、各コマンドに対する外観定義と挙動定義の生成が行われる。こうして各遷移コマンドについてグループコマンドであれば下位のコマンドが取り出され、それぞれ外観定義や挙動定義が生成される。この処理は図8ステップS5と同一である。そして、これらの処理が全コマンドの遷移線について終了すると、ステップS6に移り、グループコマンドの展開が完全かどうかを判断する。全てのグループコマンドについて最下位のコマンドまで展開して外観定義や挙動定義の生成が完了していればこの処理を終了する。もし、展開がされていないものがあれば、ステップS3に移りデフォルトの処理を行う。即ち、先に説明したように、ウィンドウ遷移を伴わない内部処理用のコマンドについてその挙動定義等を行う。

【0032】〈具体例3の効果〉以上のような具体例によって、複数のコマンドが一体に取り扱われるような処理の場合に、これらのコマンドを一括してグループ名で表示し、グループ名によって指定を行うことによって、一挙にその外観定義や挙動定義の生成を行うことができる。従って、複数のコマンドの形や形状、配置等も定型化され、更にウィンドウの生成処理が簡便になる。なお、上記のようにグループ化されたコマンドはウィンドウ遷移のためのものだけでなく、内部処理のものと同様に一括生成の対象にすることが可能である。

【0033】〈具体例4〉ウィンドウを用いたユーザインタフェースでは、所定の処理や情報を選択する場合に、メニュー構造のリストが頻繁に使用される。この場合、各メニュー項目の内容自体はそれぞれ相違しても、その階層構造は比較的定型的なものが多い。従って、このようなメニュー階層構造についてのも具体例により定型化し一括登録を可能にする。

【0034】図12に、メニュー階層構造の例(その1)を示す。メニュー階層構造は、例えばこの図に示すように構成される。即ち、メニュー項目M1の下にM11、M12が存在し、M11の下にM111、M112が存在する。即ち、M11を指定すると、M11とM1

2のいずれかの選択が要求され、M11を選択すればM111とM112のいずれかの選択が要求されるといった構成となっている。このような各メニュー項目M1～M212に対応させてコマンド名を定義し、更にそのコマンドの階層構造上の位置付けやキャプションを定義して、図2に示したような形式のコマンド辞書を用意する。

【0035】図13には、このようなメニュー階層構造を含むコマンド辞書の例を示す。上位のメニュー項目から下位のメニュー項目に移る度に2種類のメニュー項目に分割されるような図12に示したメニュー階層構造は、このコマンド辞書に示すようにして定義することができ、プログラマーがメニュー階層構造をウィンドウ上に設定する場合には、このうちの任意のメニュー項目を指定して利用する。なお、このコマンド辞書において、親コマンドとあるのは、そのコマンドの親に相当するコマンド名のことである。また、位置情報というのは、メニュー階層構造のどの場所にこのコマンドが存在するかを示す情報である。キャプションは、図12に示したような各メニュー項目に対応する。

【0036】図14に、別のメニュー構造例を示す。この例で最終的に必要とするコマンドは、M1111、M1121、M222の3種類である。これらのコマンドのみを必要とする場合のメニュー構造は、この図14に示すようになり、他の部分は削除される。

【0037】図15に、メニュー構造例(その3)を示す。例えばM112だけを必要とするウィンドウのメニュー構造はこのようなになる。このメニュー構造は、M1、M11、M112だけで構成され、それ以外のコマンドは削除されている。即ち、これだけのメニュー構造が必要であれば、コマンド辞書に示した他の各メニュー項目は表示されず、あるいは不活性状態で表示される。不活性状態というのは、クリックしても動作しない薄い字で表示されるような状態をいう。

【0038】図16には、メニュー階層構造を使用するダイアログフローの説明図を示す。この図において、ウィンドウW1が表示されている場合に、メニュー111、即ち項目名がM1111のコマンドが実行されると、ウィンドウW2に遷移する。また、ウィンドウW1が表示されている場合に、メニュー121が実行されるとウィンドウW3が表示される。ウィンドウW1が表示されている場合に、メニュー222が実行されると、ウィンドウW4に遷移する。ウィンドウW2を表示している場合に、メニュー112が実行されると、ウィンドウW5が表示される。ウィンドウW1からウィンドウW2、W3、W4に遷移する場合のメニュー構造は、図14を用いて説明したものである。また、ウィンドウW2からウィンドウW5に遷移する場合のメニュー構造は図15に示したものである。

【0039】このようなメニュー構造に従ってコマンド

11

を選択し、ウィンドウを遷移させるような場合、メニュー構造を表示させ、該当するメニュー項目についてウィンドウを遷移させるための挙動を定型化しておけば、ウィンドウ生成処理が容易になる。この具体例では、そのような処理をこれまでの具体例と同様にして自動化する。

【0040】図17は、メニュー階層構造を使用したダイアログフローの動作フローチャートである。この図のステップS1、S2は、これまで説明した図8や図11に示したステップS1、S2と全く同様の処理を示す。図17のステップS3は、メニュー階層構造かどうかを判断する処理である。ここで、メニュー階層構造であれば、ステップS4において親メニューが無いメニュー、即ち最上位のメニューに到達するまで親メニューの記録を行うことによってそのメニュー項目の位置付けを明確にする。更に、ステップS5に移り、該当するコマンドに対する外観定義と挙動定義の生成を行う。この処理の内容は、図11に示したステップS5の処理と同様である。

【0041】図16に示した全ての遷移線に対応してステップS2からステップS5の処理を完了すると、ステップS6に移り、メニュー項目の順序を位置情報により定める。即ち、同一の親メニューを持つメニュー項目が複数ある場合に、その項目同士の位置情報を比べて、例えば位置情報の小さいものを先に表示するように並べ替えを行う。

【0042】〈具体例4の効果〉これらによって、メニュー階層構造のコマンドを所定の順に表示し選択させる処理についても、メニュー階層構造上のコマンドに対応する処理を外観の定義や挙動定義と共に登録し、これまでの具体例と同様に定型化されたユーザインタフェースが自動的に生成できる。従って、例えば図9や図10で示したように、ウィンドウW1上で“確認”というボタンをプッシュすると“了解”と“取消し”というメニューが表示され、更に“取消し”というメニューを選択すると、その下層の“再試行”、“取消し終了”といったメニューが表示されるような挙動が一括生成される。

【0043】〈具体例5〉この具体例では、ウィンドウ上に配置される複数のボタンのボタン同士の位置関係、あるいはどのボタンをデフォルトボタンにするかキャンセルボタンにするかといった基準を定め、ユーザインタフェースの定型化を図るようになっている。

【0044】図18には、このようなプッシュボタンの配置等を規定したコマンド辞書の例説明図を示す。この図において、位置情報というのは、ボタンを配置する順を定める情報である。即ち、ウィンドウ上で上から下にあるいは左から右に順にボタンが並べて配列される場合には、この数値が少ないほど左あるいは上にボタンが配置されることになる。従って、位置情報が“3”のものと、位置情報が“8”のボタンが表示される場合には、

12

位置情報が“3”のボタンが左側に表示される。また、デフォルト優先度は、この数値が小さいものほど優先してデフォルトボタンの設定が行われる。即ち、複数のボタンが表示されていた場合に、デフォルト優先度が最も小さい数値のボタンがリターンキーを押した場合に直ちに実行されるボタンとなる。キャンセル優先度も同様で、最も数値の少ないものがキャンセルボタン属性を与えられる。なお、例えばこのようなデフォルト優先度やキャンセル優先度に“0”という値を設定した場合、そのプッシュボタンにはこのような属性を与えないという意味に使うことができる。

【0045】図19には、図18に示した5つのB1～B5までのボタンをウィンドウ6上に配置した例を示す。この場合、図18に示した位置情報は、B1、B4、B3、B5、B2の順に大きくなる。従って、各ボタンは図19に示すように、この位置情報順に並べられる。また、最小のデフォルト優先度を持つものは、B1のボタンであって、最小のキャンセル優先度を持つものはB2のボタンである。そこで、B1のボタンがデフォルトボタンとなり、B2のボタンがキャンセルボタンとなる。

【0046】図20には、別のボタン配置結果例説明図を示す。この図では、3個のボタンB4、B1、B5をウィンドウ6上に表示した。この場合には位置情報がB4、B3、B5の順に大きくなるため、この順にボタンが配置されている。また、キャンセルボタンはB4、デフォルトボタンはB5という設定がなされる。このようにデフォルトボタンの設定やキャンセルボタンの設定、ボタンの並び順等も予めコマンド辞書に設定しておくことによって、プログラマーが異なっても同様の思想で違和感の無いユーザインタフェースが生成できる。なお、ここではデフォルトボタンの設定やキャンセルボタンの設定のことを、デフォルト設定と呼ぶことにする。

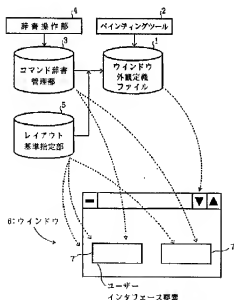
【0047】図21にはこのボタン情報を使用した外観挙動の定義生成手順フローチャートを示す。図のステップS1、S2は、図5で説明したステップS1、S2と同様の処理である。また、ステップS3では、コマンドがボタンかどうかを判断する。そして、ステップS4において、ボタンの場合には位置情報やデフォルト優先度、キャンセル優先度をコマンド辞書を参照して記録する。そして、ステップS5において、コマンドに対する処理、即ちそのコマンドに対する外観定義の生成を行う。この処理は図5ステップS4等と同様の処理である。後はこれまでの例と同様にステップS2～ステップS5の処理を繰り返して、全てのコマンドについての処理が終了すると、ステップS6に移り、ボタン位置関係の決定を行う。その決定手順は、既に図19や図20を用いて説明した通りである。そして、ステップS7において、デフォルトボタン、キャンセルボタンを決定し、処理を終了する。

【0048】〈効果〉以上により、この具体例においてもボタンの配列順まで標準化された品質の高いユーザーインタフェースを高い生産性で構築することができる。即ち、上記のような各具体例によって、ユーザーインタフェース構成要素をウィンドウ上に配置する際に必要な定型化された情報が、コマンド辞書等を参照することにより取り出され、定型化された基準でユーザーインタフェースを生成することができる。また、ウィンドウの外観のみならずボタンの形状、配置やウィンドウの遷移関係についても一定の標準化されたものが生成できるため、違和感の無い操作性のよいユーザーインタフェースが得られる。

【図面の簡単な説明】

【図1】本発明のユーザーインタフェース生成装置具体例を示す概略図である。
 【図2】コマンド辞書の内容説明図である。
 【図3】レイアウト基準の内容説明図である。
 【図4】ウィンドウ名とコマンドの対応表説明図である。
 【図5】本発明によるユーザーインタフェース生成動作説明図である。
 【図6】本発明のユーザーインタフェース生成装置概略図である。
 【図7】ダイアログフローの内容説明図である。
 【図8】ダイアログフロー指定部を利用した本発明の装置の動作フローチャートである。
 【図9】グループ機能付きコマンド辞書の内容説明図で

【図1】



本発明のユーザーインタフェース生成装置概略図（その1）

ある。

【図10】コマンドグループを利用したダイアログフローの説明図である。

【図11】コマンドグループを使用したダイアログフロー指定部利用の動作フローチャートである。

【図12】メニュー構造例説明図（その1）である。

【図13】メニュー階層構造を含むコマンド辞書の例説明図である。

【図14】メニュー構造例説明図（その2）である。

【図15】メニュー構造例説明図（その3）である。

【図16】メニュー階層構造を利用するダイアログフロー説明図である。

【図17】メニュー階層構造を利用したダイアログフローの動作フローチャートである。

【図18】ボタン配置を定めたコマンド辞書の例説明図である。

【図19】ボタン配置結果例（その1）である。

【図20】ボタン配置結果例（その2）である。

【図21】ボタン情報を使った外観自動定義生成手順動作フローチャートである。

【符号の説明】

- 1 ウィンドウ外観定義ファイル
- 2 コマンド辞書管理部
- 3 レイアウト基準指定部
- 4 ウィンドウ
- 5 ユーザーインタフェース要素

【図2】

コマンド名	コマンド組別	属性		
		キャプション	ショートカット	大きさ
開く	メニュー	"開く"	Q	...
検索	内部イベント	—	—	...
了解	プッシュボタン	"OK"	—	300×500...
...

コマンド辞書の内容説明図

【図3】

プッシュボタン配置	上下から右下へ等間隔に
...	...
...	...

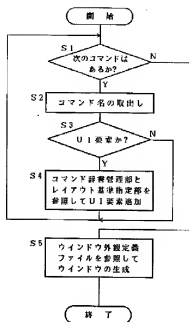
レイアウト基準の内容説明図

【図4】

ウィンドウ名	コマンド名
W 1	置く
W 1	移動
W 2	内部イベント

ウィンドウ名とコマンドの対応表説明図

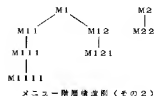
【図5】



UI:ユーザーインタフェース

本発明によるユーザーインタフェース生成動作

【図14】



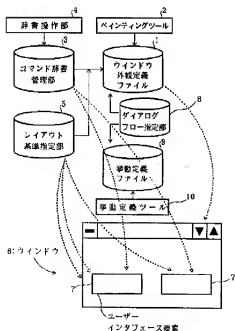
メニュー階層構造例(その2)

【図15】



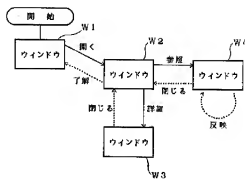
メニュー階層構造例(その3)

【図6】



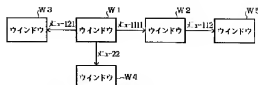
本発明のユーザーインタフェース生成装置概略図(その2)

【図7】



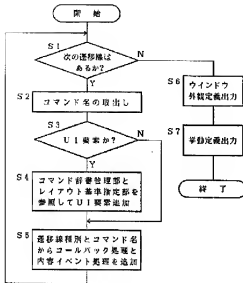
ダイアログフローの内容説明図

【図16】



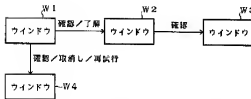
メニュー階層構造を使用したダイアログフロー

【図8】



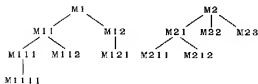
ダイアログフロー指定部利用の動作

【図10】



コマンドグループを利用したダイアログフロー説明図

【図12】



メニュー階層構造例(その1)

【図9】

コマンドグループ名	構成コマンド
確認	了解+取消し
取消し	再試行+取消し終了

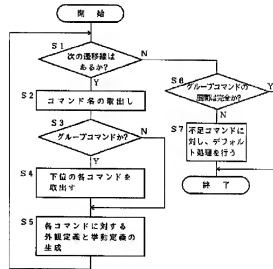
(a)

コマンド名	コマンド種類	属性		
		キャプション	ニコニコ	大きさ
了解	プッシュボタン	"OK"	a	100×20
再試行	プッシュボタン	"RETRY"	e	100×20
取消し終了	プッシュボタン	"CANCEL"	c	100×20

(b)

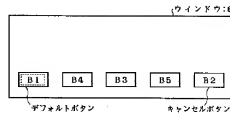
グループ渡継付きコマンド定義説明図

【図11】



ダイアログフロー指定部利用の動作

【図19】



デフォルトボタン

キャンセルボタン

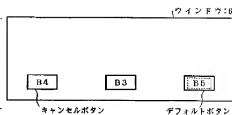
ボタン配置結果例(その1)

【図13】

コマンド名	コマンド種別	親コマンド	位置情報	キャプション	他の属性
メニュー1	メニュー	—	3	M1	
メニュー11	メニュー	メニュー1	1	M11	
メニュー111	メニュー	メニュー11	5	M111	
メニュー1111	メニュー	メニュー111	2	M1111	
メニュー112	メニュー	メニュー11	10	M112	
メニュー12	メニュー	メニュー1	3	M12	
メニュー121	メニュー	メニュー12	10	M121	
メニュー2	メニュー	—	5	M2	
メニュー21	メニュー	メニュー2	1	M21	
メニュー211	メニュー	メニュー21	2	M211	
メニュー212	メニュー	メニュー21	4	M212	
メニュー22	メニュー	メニュー2	3	M22	
メニュー23	メニュー	メニュー2	5	M23	

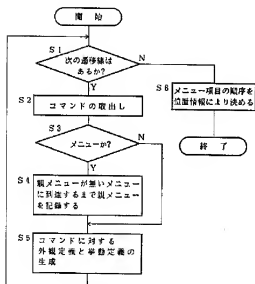
メニュー階層構造を含むコマンド屏幕の例

【図20】



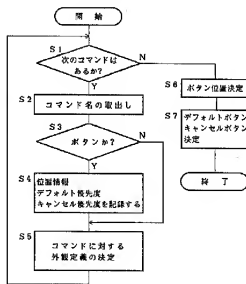
ボタン配置画面例(その2)

【図17】



メニュー階層構造を使用したダイアログフロー利用の動作

【図21】



ボタン情報を使った外観挙動定義生成手順

【図18】

コマンド名	コマンド種別	位置情報	デフォルト優先度	キャンセル優先度	他の属性
B1	ブッシュボタン	3	5	0	
B2	ブッシュボタン	30	0	4	
B3	ブッシュボタン	10	0	8	
B4	ブッシュボタン	8	0	10	
B5	ブッシュボタン	11	7	8	

ボタン配置を定めたコマンド群書の例